IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Application Serial No. 10/686,316
Filing Date ..... Oct 15, 2003
Inventor ..... Peter L. Montgomery
Group Art Unit 2131
Examiner ..... SHIN HON CHEN
Attorney's Docket No. MS1-1648US
Confirmation No. 8266
Title: Utilizing SIMD Instructions Within Montgomery Multiplication

To: The Honorable Commissioner for Patents
Mail Stop Appeal Brief- Patents
PO Box 1450
Alexandria, Virginia 22313-1450

From: Beatrice L. Koempel-Thomas (Tel. 509-324-9256 x259; Fax 509-323-8979)
Customer No. 22801

BRIEF OF APPELLANT

The Applicant has filed a timely Notice of Appeal from the action of the Examiner in finally rejecting all of the claims that were considered in this application. This Brief is being filed under the provisions of 37 C.F.R. § 41.37. The Filing Fee, as set forth in 37 C.F.R. § 41.20(b)(2) and the appropriate forms accompany this Brief for payment of any additional fees (e.g., for a two-month extension).

TABLE OF CONTENTS

I. REAL PARTY IN INTEREST

The real party in interest is Microsoft Corporation, by way of assignment from Peter L. Montgomery who is the named inventive entity and is captioned in the present brief.

II. RELATED APPEALS AND INTERFERENCES

None.

III. STATUS OF CLAIMS

Claims 1-4, 7, 8, 10, 12, and 14-22 are rejected. The rejections of claims 1-4, 7, 8, 10, 12, and 14-22 are the subject of this appeal.

The history of the claims is as follows:

a. Claims 1-24 were originally filed.

b. In an Office Action mailed March 27, 2007, claims 1-24 were rejected under 35 U.S.C. § 101 as being directed to non-statutory subject matter as "recit[ing] a method of 'performing Montgomery multiplication' and the claim limitations seem[ing] to be directed to an abstract idea without limitation to a practical application because the claim merely recites process of mathematical formula without producing tangible result"; claims 1-5, 7-10, 12, 15-21, and 23 were rejected under 35 U.S.C. 102(b) as anticipated by Posch et al. "RNS-Modulo Reduction Upon a Restricted Base Value Set and its Applicability to RSA Cryptography"; claims 14 and 22 were rejected under §103(a) as being unpatentable over Posch; and claims 6, 11, 13, and 24 were objected to as being dependent upon a rejected base claim, but allowable if all of the limitations of the base claim and any intervening claims were incorporated and written to overcome the rejections under 35 U.S.C. § 101.

c. An Examiner Interview was held on June 20, 2007.

d. An Office Action Response was filed on June 27, 2007 where Applicant amended claims 1, 7, 12, 15, and 20 and canceled claims 5, 6, 9, 11, 13, 23, and 24.

e. In a Final Action mailed August 29, 2007, claims 1-4, 7, 8, 10, 12, and 14-22 were rejected under 35 U.S.C. § 101 as being directed to non-statutory subject matter based on *Gottschalk v. Benson*, 409 U.S. 63 (1972).

f. An Office Action Response was filed on November 29, 2007 where Applicant did not amend any claims.

g. An Advisory Action was mailed on December 26, 2007.

h. Appellant filed a Notice of Appeal on February 28, 2008.

IV. STATUS OF AMENDMENTS

Amendments submitted in the Office Action Response filed June 27, 2007 have been entered.

V. SUMMARY OF THE CLAIMED SUBJECT MATTER

Performing many multiplication and division operations almost instantaneously, for example in security functions, is an area that taxes computing technology despite other advances in the technology. Prior to this invention the state-of-the-art of rigorous operations employing complex cryptographic algorithms used exponentiation operations involving many modular multiplications and operations on particular computer architectures. Specifically, some computer architectures supported execution of Single Instruction, Multiple Data (SIMD) where each instruction performs one operation on multiple sets of data—the same operation on each set.

Streaming SIMD Extensions 2 (SSE2) is one form of SIMD instructions that the computer architecture Pentium® 4 microprocessor, from Intel Corporation, is capable of executing. Generally, SSE2 instructions reduce the overall number of instructions used to execute a particular program task to increase overall performance. Montgomery multiplication is an algorithm for modular multiplication which avoids division by reducing double-length products from the right rather than from the left. *See Application, Page 2, paragraph [0002] to Page 3, paragraph [0005].*

Accordingly, this system and related methods for implementing Montgomery multiplication on a computer system that supports SIMD instructions was invented. One of the claimed aspects is a method of implementing the machine-level operations utilized by Montgomery multiplication, wherein the operations are performed using SSE2 (Streaming

SIMD Extensions 2) instructions. Thus, the claimed invention provides an architecture for utilizing SIMD instructions within Montgomery multiplication operating on two independent operands. *See Application, Page 3, paragraph [0006] and Page 4, paragraph [00011].*

For example, Fig. 3 is a flowchart illustrating an exemplary process of using SIMD instructions within Montgomery multiplication. A computer architecture initializes values to be used in computing the function montmul(*A, B*). More specifically, the values include integer inputs *A, B*, temporary arrays *T1, T2*, and modulus *N*. The computer architecture sets temporary arrays *T1, T2* to zero. Values for input *B* and modulus *N* are interleaved into a vector, and a pair of elements from these arrays will fit in one of the registers of the computer architecture.

An iterative loop begins to process the digits of input *A*, from right-to-left. Assuming that array *T1* is to be updated with a multiple of input *B* (with multiplier *mul1* being a digit from array *A*), the processor determines what multiple *mul2* of modulus *N* allows the update arrays *T1, T2* to end with the same digit(s). Multiplication (digit of *A*) times *B* (i.e., *mul1*B[i]*) and multiplication of the determined multiple times modulus N (i.e., mul2*N[i]) are performed. The arrays *T1, T2* are updated. By using two arrays *T1* and *T2*, the multiplication *AB* is interleaved with the multiplication *qN* in the Montgomery multiplication. The loop continues until all digits in input *A* have been processed, and the result $T1 - T2 \pmod{N}$ is returned. The claimed architecture, using SIMD instructions within Montgomery multiplication updates arrays *T1* and *T2* together. When (*T1[i], T2[i]*) is referenced from

memory, the operands are in adjacent locations, as this is how they were stored on a previous iteration. More precisely, they are stored as adjacent 64-bit locations, even though their values fit into 32 bits. Pairs like (*B[i], N[i]*) are copied to adjacent locations early, and reused as each digit of *A* is processed *See Application, Page 18, paragraph [00060] to Page 26, paragraph [00085].*

Independent Claim 1 recites a computer system comprising:

- a memory wherein results of processing are stored; and (e.g., reference numbers 108(1), 108(2) … 108(M), Fig. 1; page 4, paragraphs [00013] and [00014]);
- a processor that supports SIMD instructions, the processor being configured to perform Montgomery multiplication using SIMD instructions, wherein the Montgomery multiplication has a loop of instructions, and each iteration of the loop involves, excepting copy operations, using no more than eight SIMD instructions and wherein the SIMD instructions comprise two load instructions, one multiply instruction, two add instructions, one copy instruction, one bitwise AND instruction, one store instruction, and one shift instruction (e.g., reference number 102, Fig. 1, page 4, paragraphs [00013] and [00014], page 5, paragraph [00015]; Fig. 2, pages 16-17, paragraphs [00050] – [00051]; and page 34, original claims 5, and 6).

Independent Claim 7 recites a processing system comprising:

- a processor having a set of registers, the processor being configured to support SIMD instructions; and (e.g., reference number 102, Fig. 1, page 4, paragraphs [00013] and [00014], page 5, paragraph [00015]; Fig. 2, pages 16-17, and paragraphs [00050] – [00051]); and
- a set of SIMD instructions, executable by the processor, to perform Montgomery multiplication (e.g., reference number 110, Fig. 1, page 5, paragraph [00016]):
    - montmul(A, B) = rem($(AB - qN)/R$, N), where q = rem($AB\ N'$, R) where A and B are integers, q is a quotient, N is a modulus, R is an integer that is coprime to modulus N, and N' is an integer such that $N\,N' \equiv 1$ (mod R), wherein the integer B and the modulus N are implemented as arrays, and at least one SIMD instruction is used to update a first array T_1 with multiples of B for computing AB and to update a second array T_2 with multiples of N for computing qN, wherein a first register holds elements of the B and N arrays (e.g., pages 5-12, paragraphs [00018] – [00042]; Fig. 2, pages 16-17, paragraphs [00050] – [00051]; page 34, original claim 7; and page 35, original claims 9 and 11);
- a second register holds an element of the first array T_1 and an element of the second array T_2; (e.g., Fig. 2, pages 16-17, paragraphs [00050] – [00051]; and page 35, original claim 11); and

- a third register is used to hold results of the first array T_1 being updated with a multiple of B and the second array T_2 being updated with multiples of N (e.g., Fig. 2, pages 16-17, paragraphs [00050] – [00051]; and page 35, original claim 11).

Independent claim 12 recites a computer readable medium comprising computer-executable SIMD instructions that, when executed, direct a processor to perform Montgomery multiplication, the instructions comprising:

- a first SIMD instruction to load elements of arrays B and N into a first register (e.g., Fig. 2; page 16, paragraph [00050]);
- a second SIMD instruction to load elements of arrays T_1 and T_2 into a second register (e.g., Fig. 2; page 16, paragraph [00050]);
- a third SIMD instruction to multiply an element in the array B by a first multiple and an element in the array N by a second multiple (e.g., Fig. 2; page 16, paragraph [00051]);
- fourth and fifth SIMD instructions to add results of the third SIMD instruction to the array elements loaded by the second SIMD instruction and to any carries saved from a previous iteration (e.g., Fig. 2; page 16, paragraph [00051]);
- sixth and seventh SIMD instructions to separate each output of the fifth SIMD instruction into two reduced size results, one that fits into the arrays T_1 and T_2 and

another that represents a carry for a next iteration (e.g., Fig. 2; pages 16-17, paragraph [00051]);

- an eighth SIMD instruction to update an element of array T_1 and an element of array T_2, in memory (e.g., Fig. 2; pages 16-17, paragraph [00051]); and
- an instruction to store the result of the final iteration (e.g., Fig. 2; pages 16-17, paragraph [00051]).

Independent Claim 15 recites a method for computing Montgomery multiplication, whereby Montgomery multiplication is performed within a cryptographic function in a computer (e.g., Fig. 2; page 5, paragraph [00016]; pages 9-10, paragraph [00033]), the method comprising:

- montmul(A, B) = rem($(AB - qN)/R$, N), where $q = \text{rem}(AB\,N', R)$ where A and B are integers, q is a quotient, N is a modulus, R is an integer that is coprime to modulus N, and N' is an integer such that $N\,N' \equiv 1 \pmod{R}$ (e.g., page 7, paragraphs [00021] and [00023]), the method comprising:
  - iteratively performing, for each digit of integer A from right to left(e.g., page 19, paragraph [00062]):
    - with array T_1 being updated by a product of input B times the digit of integer A, determining what multiple of modulus N allows the updated arrays T_1, T_2 to end with the same digit (e.g., page 19, paragraph [00062]);

- multiplying the input B by the digit of integer A and multiplying the modulus N by the determined multiple (e.g., page 19, paragraph [00062]); and
- updating the arrays T_1, T_2 (e.g., page 19, paragraph [00062])

- storing the result of the final iteration (e.g., page 19, paragraph [00062]).

Independent Claim 20 recites a method whereby Montgomery multiplication is performed within a cryptographic function in a computer (e.g., Fig. 2; page 5, paragraph [00016]; pages 9-10, paragraph [00033]), the method comprising:

- initializing a set of registers with values used in performing Montgomery multiplication (e.g., page 19, paragraph [00061]);
- computing the Montgomery multiplication with SIMD instructions on the values stored in the registers, wherein the Montgomery multiplication has a loop of instructions, and each iteration of the loop is performed using not more than nine SIMD instructions wherein the nine SIMD instructions comprise (e.g., Fig. 2; pages 16-17, paragraphs [00050] and [00051]):

- two load instructions (e.g., Fig. 2; page 16, paragraph [00050]);
- one multiply instruction (e.g., Fig. 2; page 16, paragraph [00051]);
- two add instructions (e.g., Fig. 2; page 16, paragraph [00051]);
- one copy instruction (e.g., Fig. 2; page 16, paragraph [00051]);
- one bitwise AND instruction (e.g., Fig. 2; page 16, paragraph [00051]);
- one store instruction (e.g., Fig. 2; page 16, paragraph [00051]); and
- one shift instruction (e.g., Fig. 2; pages 16-17, paragraph [00051]); and

- storing the result of the final iteration of the loop (e.g., Fig. 2; pages 16-17, paragraph [00051]; page 19, paragraph [00062]).

VI. GROUND OF REJECTION TO BE REVIEWED ON APPEAL

Appellant respectfully requests that the Board review the ground, as stated by the Examiner, for rejection of all pending claims (1-4, 7, 8, 10, 12, and 14-22) in the instant application as being directed to non-statutory subject matter under 35 U.S.C. § 101.

The only ground of rejection is disputed as applied to all claims:

The issue in dispute is whether the claims are non-statutory; the Examiner asserts that all seventeen pending claims (1-4, 7, 8, 10, 12, and 14-22) are directed to non-statutory subject matter based on the holding of *Gottschalk v. Benson*, 409 U.S. 63 (1972), (hereinafter "*Gottschalk*"). Appellant disagrees. Further, the evidence is insufficient to support the Examiner's finding. Appellant requests a decision on this issue. In particular, Appellant requests review of the ground of rejection, based on specific evidence and argument supplied by the Examiner, which are of record in the Final Office Action of (August 29, 2007).

VII. ARGUMENT

Appellant disputes the ground of rejection. Appellant submits that the Office erred in rejecting all seventeen pending claims (1-4, 7, 8, 10, 12, and 14-22) under 35 U.S.C. § 101 based on the holding of *Gottschalk* because the evidence does not support such rejection of the claims. Appellant requests that the Board review the Examiner's ground for rejection of all seventeen claims (1-4, 7, 8, 10, 12, and 14-22) in the instant application as being directed to non-statutory subject matter based on *Gottschalk*, under 35 U.S.C. § 101.

Specific Errors

(a) Independent Claims. Appellant submits that the Office erred in rejecting claims 1, 7, 12, 15, and 20 under 35 U.S.C. § 101 because the holding of *Gottschalk* does not foreclose all claims including mathematical algorithms. The claims at issue in *Gottschalk* were ultimately rejected not because they included mathematical algorithms, but because "[t]he claims were not limited to any particular art or technology, to any particular apparatus or machinery, or to any particular end use." *Gottschalk v. Benson*, 409 U.S. 63, 64 (1972). The independent claims on appeal contrast with the claims at issue in *Gottschalk* for at least the reasons presented below. Appellant addresses arguments to the independent claims representing all 17 claims (1-4, 7, 8, 10, 12, and 14-22) in groups, with claim 1 being exemplary of claims 1-10, claim 12 being exemplary of claims 12-14, and claim 15 being exemplary of claims 15-22.

Standards

All claims are rejected under 35 U.S.C. § 101:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

In the 1952 Patent Act, the Congress expressly defined "process" when substituting "process" for "art" as one of the four categories of patent eligible subject matter under 35 U.S.C. § 101. "The term 'process' means process, art, or method, and includes a new use of a known process, machine, manufacture, composition of matter, or material," 35 U.S.C. § 100 (b) (2008).

The determination of whether claims are directed to patent eligible subject matter is further based on several prominent Supreme Court and Federal Circuit Court of Appeals decisions. Analysis begins with determining whether each claim falls within one of the four enumerated categories of patentable subject matter recited in 35 U.S.C. § 101 (i.e., process, machine, manufacture, or composition of matter). However, the analysis does not end there because case law states that claims directed to nothing more than abstract ideas (such as mathematical algorithms), natural phenomena, and laws of nature are not eligible for patent protection. *Diamond v. Diehr*, 450 U.S. 175, 185 (1981); accord, e.g., *Diamond v. Chakrabarty*, 447 U.S. 303, 309 (1980); *Parker v. Flook*, 437 U.S. 584, 589 (1978);

Gottschalk v. Benson, 409 U.S. 63, 67-68 (1972) (emphasis added).

While abstract ideas, natural phenomena, and laws of nature are not eligible for patenting, "a process, machine, manufacture, or composition of matter employing a law of nature, natural phenomenon, or abstract idea is patentable subject matter even though a law of nature, natural phenomenon, or abstract idea would not, by itself, be entitled to such protection." *State Street Bank & Trust Co. v. Signature Financial Group, Inc.*, 149 F.3d 1368, 1374 (Fed. Cir. 1998) (emphasis added).

"Transformation and reduction of an article 'to a different state or thing' is [a] clue to the patentability of a process claim that does not include particular machines." *Diehr*, 450 U.S. at 183 (quoting *Gottschalk*, 409 U.S. at 70) (emphasis added).

In evaluating whether a claim meets the requirements of section 101, each claim must be considered as a whole to determine whether it is for a particular application of an abstract idea, natural phenomenon, or law of nature, and not for the abstract idea, natural phenomenon, or law of nature itself. MPEP 2106 (C) ("Determine Whether the Claimed Invention Falls Within 35 U.S.C. 101 Judicial Exceptions - Laws of Nature, Natural Phenomena and Abstract Ideas", discussing e.g., *Diehr*, 450 U.S. at 185, 209 USPQ at 7; accord, e.g., *Chakrabarty*, 447 U.S. at 309, 206 USPQ at 197; *Parker v. Flook*, 437 U.S. 584, 589, 198 USPQ 193, 197 (1978); *Benson*, 409 U.S. at 67-68 , 175 USPQ at 675; *Funk*, 333 U.S. at 130, 76 USPQ at 281; *O'Reilly v. Morse*, 56 U.S. (15 How.) 62, 113-114 (1853)).

Although there is some tension between the principles enunciated by the cases included in this group, Appellant maintains that the instant claims on appeal meet the requirements of 35 U.S.C. § 101 as patent eligible subject matter when examined through the combined lens of these cases.

(a) Analysis of the Office's Rejection of the Independent Claims

Claims 1-10 as a Group represented by Independent Claim 1:

For purposes of expediency, Appellant discusses the errors in the rejection of the independent claims 1 and 7 directed to specifically: "a computer system," claim 1; and "a processing system," claim 7, by presenting claim 1 as an example. Claim 1 recites (emphasis of the system in bold):

*A **computer system** comprising:*

*a **memory wherein results of processing are stored**; and*

*a processor that supports SIMD instructions, the **processor being configured to perform Montgomery multiplication using SIMD instructions**, wherein the Montgomery multiplication has a loop of instructions, and each iteration of the loop involves, excepting copy operations, using no more than eight SIMD instructions and wherein the SIMD instructions comprise two load instructions, one multiply instruction, two add instructions, one copy instruction, one bitwise AND instruction, one store instruction, and one shift instruction.*

Claims 12-14 as a Group represented by Independent Claim 12:

Appellant discusses the errors in the rejection of independent claim 12 directed to specifically: "a computer readable medium." Claim 12 recites (emphasis of the computer readable medium in bold):

A ***computer readable medium*** *comprising* ***computer-executable SIMD instructions that, when executed, direct a processor to perform Montgomery multiplication****, the instructions comprising:*

a first SIMD instruction to load elements of arrays B and N into a first register;

a second SIMD instruction to load elements of arrays T_1 *and* T_2 *into a second register;*

a third SIMD instruction to multiply an element in the array B by a first multiple and an element in the array N by a second multiple;

fourth and fifth SIMD instructions to add results of the third SIMD instruction to the array elements loaded by the second SIMD instruction and to any carries saved from a previous iteration;

sixth and seventh SIMD instructions to separate each output of the fifth SIMD instruction into two reduced size results, one that fits into the arrays T_1 *and* T_2 *and another that represents a carry for a next iteration;*

an eighth SIMD instruction to update an element of array T_1 and an element of array T_2, in memory; and

an instruction to store the result of the final iteration.

Claims 15-24 as a Group represented by Independent Claim 15:

For purposes of expediency, Appellant discusses the errors in the rejection of the independent claims 15 and 20 directed to specifically: "a method for computing Montgomery multiplication, whereby Montgomery multiplication is performed within a cryptographic function," claim 15; and "a method whereby Montgomery multiplication is performed within a cryptographic function," claim 20, by presenting claim 15 as an example. Claim 15 recites (emphasis of method being performed within a cryptographic function in a computer in bold):

*A **method for computing Montgomery multiplication**, whereby **Montgomery multiplication is performed within a cryptographic function in a computer**, the method comprising:*

montmul(A, B) = rem((AB − qN)/R, N), where q = rem(AB N′, R)

where A and B are integers, q is a quotient, N is a modulus, R is an integer that is coprime to modulus N, and N′ is an integer such that N N′ ≡ 1 (mod R), the method comprising:

iteratively performing, for each digit of integer A from right to left:

with array T_1 being updated by a product of input B times the digit of integer A, determining what multiple of modulus N allows the updated arrays T_1, T_2 to end with the same digit;

multiplying the input B by the digit of integer A and multiplying the modulus N by the determined multiple; and

updating the arrays T_1, T_2

storing the result of the final iteration.

Whereas the claims ultimately rejected in *Gottschalk* were to methods that were not limited to a particular technology, apparatus, or end use, Applicant submits that claims 1, 7, and 12 are directed to physical machines producing useful, concrete, and tangible results—not disembodied mathematical concepts. *See infra In re Alappat*, 33 F.3d 1526 (Fed. Cir. 1994) (en banc).

Because of at least these recitations, the instant claims are substantively different than the claims in *Gottschalk*. Claims 1, 7, and 12 would not "in practical effect . . . be a patent on the algorithm itself." *See Gottschalk* 409 U.S. at 71.

Whereas the claims ultimately rejected in *Gottschalk* were to methods that were not limited to a particular technology, apparatus, or end use, Applicant maintains that claims 15 and 20, directed to methods, are limited to a particular technology, namely cryptography, and therefore the instant claims would not "wholly pre-empt the mathematical formula."

Because of at least these recitations, the instant claims are substantively different than the claims in *Gottschalk*. Claims 15 and 20 would not "in practical effect . . . be a patent on the algorithm itself." *See Gottschalk* 409 U.S. at 71.

Stated Grounds of Rejection of all Claims

In the Final Office Action (8/29/07, p. 2), the Office stated the following grounds for rejection of all pending claims:

> Claims 1-4, 7, 8, 10, 12, and 14-22 rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.
>
> This application sought to patent a method of performing Montgomery Multiplication. A procedure for solving a given type of mathematical problem is known as an algorithm. The procedures set forth in the present claims are of that kind; that is to say, they are a generalized formulation for problems to solve mathematical problems of calculating multiplication. The mathematical procedures can be carried out in existing computers long in use, no new machinery being necessary. And, as noted, they can also be performed without a computer. The mathematical formula involved here has no substantial practical application except in connection with a digital computer, which means that patent would wholly pre-empt the mathematical formula and in practical effect would be a patent on the algorithm itself.
>
> *Gottschalk v. Benson.*

The Office's argument relies on the holding of *Gottschalk*. This evidence is identified and discussed below.

GROUND OF REJECTION:

Gottschalk v. Benson

The claims at issue in *Gottschalk* were ultimately rejected not because they included mathematical algorithms, but because "[t]he claims were not limited to any particular art or technology, to any particular apparatus or machinery, or to any particular end use." *Gottschalk v. Benson*, 409 U.S. 63, 64 (1972). The independent claims presented above contrast with the claims at issue in *Gottschalk* for at least the reasons presented below.

The claims above are directed to specifically:

- "a computer system," claim 1;
- "a processing system," claim 7;
- "a computer readable medium," claim 12;
- "a method for computing Montgomery multiplication, whereby Montgomery multiplication is performed within a cryptographic function in a computer," claim 15; and
- "a method whereby Montgomery multiplication is performed within a cryptographic function in a computer," claim 20.

Specific Evidence Relied on by the Office as Grounds for Rejection

The Office states:

> A procedure for solving a given type of mathematical problem is known as an "algorithm." The procedures set forth in the present claims are of that kind; that is to say, they are a generalized formulation for programs to solve mathematical problems of calculating multiplication. The mathematical procedures can be carried out in existing computers long in use, no new machinery being necessary. And, as noted, they can also be performed without a computer. The mathematical formula involved here has no substantial practical application except in connection with a digital computer, which means that [sic] patent would wholly preempt the mathematical formula and, in practical effect, would be a patent of the algorithm itself.

Final Office Action, August 29, 2007.

Previously the Office has taken the position that the claims "recit[ed] a method of 'performing Montgomery multiplication' and the claim limitations seem[ed] to be directed to an abstract idea without limitation to a practical application because the claim merely recites process of mathematical formula without producing tangible result." Office Action, March 27, 2007, p. 2.

Evidence Contrasted with the position taken by the Office

The opinion in Gottschalk states, in part:

> "The patent sought is on a method of programming a general purpose digital computer to convert signals from binary-coded decimal form into pure binary form. A procedure for solving a given type of mathematical problem is known as an "algorithm." The procedures set forth in the present claims are of that kind; that is to say, they are a generalized formulation for programs to solve mathematical problems of converting one form of numerical representation to another. From the generic formulation, programs may be developed as specific

> applications. (*Gottschalk*, 409 U.S. at 65). . . . The mathematical procedures [conversion of BCD numerals to pure binary numerals] can be carried out in existing computers long in use, no new machinery being necessary. And, as noted, they can also be performed without a computer. (*Gottschalk*, 409 U.S. at 67). . . . It is conceded that one may not patent an idea. But, in practical effect, that would be the result if the formula for converting BCD numerals to pure binary numerals were patented in this case. The mathematical formula involved here has no substantial practical application except in connection with a digital computer, which means that, if the judgment below is affirmed, the patent would wholly preempt the mathematical formula and, in practical effect, would be a patent of the algorithm itself.

(*Gottschalk*, 409 U.S. at 65, 67 and 71-72).

The Office appears to rely on particular statements from Gottschalk outside of the facts of that case and the context of the ruling. Notably, the claims in Gottschalk were found to be "a generalized formulation for programs to solve mathematical problems [from which] programs may be developed as specific applications." (*Gottschalk*, 409 U.S. at 65).

By contrast, the claims at issue in the instant appeal are apparatus claims and method claims specifically directed to performing Montgomery multiplication within cryptographic functions in a computer.

The claims of Gottschalk are contrasted with the instant method claims in the table below. Each of the method claims on appeal is specifically directed to performing Montgomery multiplication within cryptographic functions in a computer, and neither of the method claims on appeal are "generalized formulation for programs to solve mathematical problems [from which] programs may be developed as specific applications," as the claims in Gottschalk were determined to be.

Method Claims of Gottschalk	The Instant Method Claims
8. The method of converting signals from binary coded decimal form into binary which comprises the steps of (1) storing the binary coded decimal signals in a reentrant shift register, (2) shifting the signals to the right by at least three places, until there is a binary '1' in the second position of said register, (3) masking out said binary '1' in said second position of said register, (4) adding a binary '1' to the first position of said register, (5) shifting the signals to the left by two positions, (6) adding a '1' to said first position, and (7) shifting the signals to the right by at least three positions in preparation for a succeeding binary '1' in the second position of said register.	15. A method for computing Montgomery multiplication, whereby Montgomery multiplication is performed within a cryptographic function in a computer, the method comprising: montmul(A, B) = rem($(AB - qN)/R$, N), where q = rem($AB\,N'$, R). where A and B are integers, q is a quotient, N is a modulus, R is an integer that is coprime to modulus N, and N' is an integer such that $N\,N' \equiv 1 \pmod{R}$, the method comprising: iteratively performing, for each digit of integer A from right to left: with array T_1 being updated by a product of input B times the digit of integer A, determining what multiple of modulus N allows the updated arrays T_1, T_2 to end with the same digit; multiplying the input B by the digit of integer A and multiplying the modulus N by the determined multiple; and updating the arrays T_1, T_2 storing the result of the final iteration.
13. A data processing method for converting binary coded decimal number representations into binary number representations, comprising the steps of (1) testing each binary digit position '1,' beginning with the least significant binary digit position, of the most significant decimal digit representation for a binary '0' or a binary '1'; (2) if a binary '0' is detected, repeating step (1) for the next least significant binary digit position of said most significant decimal digit representation; (3) if a binary '1' is detected, adding a binary '1' at the (i+1)th and (i+3)th least significant binary digit positions of the next lesser significant decimal digit representation, and repeating step (1) for the next least significant binary digit position of said most significant decimal digit representation; (4) upon exhausting the binary digit positions of said most significant decimal digit representation, repeating steps (1) through (3) for the next lesser significant decimal digit representation as modified by the previous execution of steps (1) through (3); and (5) repeating steps (1) through (4) until the second least significant decimal digit representation has been so processed.	20. A method whereby Montgomery multiplication is performed within a cryptographic function in a computer, the method comprising: initializing a set of registers with values used in performing Montgomery multiplication; computing the Montgomery multiplication with SIMD instructions on the values stored in the registers, wherein the Montgomery multiplication has a loop of instructions, and each iteration of the loop is performed using not more than nine SIMD instructions wherein the nine SIMD instructions comprise: two load instructions; one multiply instruction; two add instructions; one copy instruction; one bitwise AND instruction; one store instruction; and one shift instruction; and storing the result of the final iteration of the loop.

Furthermore, in the response to the Office Action dated March 27, 2007, Applicant amended the claims in the instant appeal to highlight the included practical utility of returning the value at the end of the loop as described in the Specification on pages 5 and 9, for example. This highlights the practical application of Montgomery multiplication to at least cryptographic functions. Also, Montgomery multiplication has the practical application of speeding processing by reducing loop iterations as set forth in the Specification on at least page 15.

Patent eligible subject matter in claims 1-4, 7, 8, 10, 12, and 14-22 under 35 U.S.C. § 101:

Diamond v. Diehr

In *Diehr*, the Supreme Court cautioned against applications that "seek[] patent protection for that formula in the abstract." *Diehr*, 450 U.S. 175, 191 (1981). Also in *Diehr*, the Court found that the application in that case did "not seek to pre-empt the use of [an] equation," but rather sought only to "foreclose from others the use of that equation in conjunction with all of the other steps in their claimed process"). Id. at 187. In *Diehr* the Court stated that *Gottschalk v. Benson* and *Parker v. Flook* "stand for no more than [the] long-established principles" that abstract ideas and natural phenomena are not patentable. Id. at 185.

Specifically the *Diehr* Court found that the claims while employing "a well-known mathematical equation, [did] not seek to pre-empt the use of that equation." Id. at 187.

Likewise, the claims at issue in the instant application employ a mathematical equation and Applicant does not seek to pre-empt the use of that equation. Rather, Applicant seeks to *foreclose from others the use of Montgomery multiplication in conjunction with all of the other steps in the processes of claims 15 and 20.*

State Street Bank

The transformation of data through a series of mathematical calculations, by a machine, constitutes a practical application of a mathematical algorithm, formula, or calculation because it produces a useful, concrete, and tangible result.

Applicant further maintains that the instant claims represent transforming data through a series of mathematical calculations into a final result. According to *State Street Bank*, 149 F.3d 1368, 1373 (Fed. Cir. 1998), such a final result constitutes a practical application of Montgomery multiplication because it produces a useful, concrete, and tangible result. For at least this additional reason, these claims meet the requirement of a practical application of a judicial exception for mathematical algorithms.

In re Comiskey

Applicant further maintains that the instant claims are patentable because they combine the use of particular physical machines which utilize SIMD instructions within Montgomery Multiplication in the technology of cryptography. According to *In re Comiskey*,

Slip. Op. 2006-1286, (Fed. Cir. 2007), such a combination represents patentable subject matter. ("While the mere use of the machine to collect data necessary for application of the mental process may not make the claim patentable subject matter, . . . these claims in combining the use of machines with a mental process, claim patentable subject matter." Id. at 23-24 (citation omitted)). However, Claims 1 and 32 in *Comiskey* were determined not to be patentable under §101 because the "claims [did] not require a machine, and . . . [did] not describe a process of manufacture . . .Comiskey's independent claims 1 and 32 [claimed] the mental process of resolving a legal dispute between two parties by the decision of a human arbitrator." Id. at 21-22. "Comiskey's independent claims 1 and 32 [sought] to patent the use of human intelligence in and of itself." Id. at 22). Unlike Comiskey's claims 1 and 32, the instant claims require a machine or are directed to a machine or article of manufacture; *they do not seek to patent the use of human intelligence in and of itself.* For at least this additional reason, these claims meet the requirements for patentability under §101.

Patent eligible subject matter in claims 1-4, 7, 8, and 10 under 35 U.S.C. § 101:

In re Alappat

Applicant respectfully submits that pending claims 1-10 recite machines constituting statutory subject matter under 35 U.S.C. §101. Applicant sets forth the legal standard for a rejection under §101, including *In re Alappat*, 33 F.3d 1526, 31 USPQ2d 1545 (Fed. Cir.

1994) addressing a similar rejection under §101 as that of the instant rejection of claims 1-10. Through analysis of *Alappat*, Applicant will show that the Office's rejection of the instant claims stands in stark disagreement with prevailing law.

The Federal Circuit in *Alappat* held that the following computer-related apparatus claim constituted statutory subject matter under 35 U.S.C. §101:

> A rasterizer for converting vector list data representing sample magnitudes of an input waveform into anti-aliased pixel illumination intensity data to be displayed on a display means comprising:
> (a) means for determining the vertical distance between the endpoints of each of the vectors in the data list;
> (b) means for determining the elevation of a row of pixels that is spanned by the vector;
> (c) means for normalizing the vertical distance and elevation; and
> (d) means for outputting illumination intensity data as a predetermined function of the normalized vertical distance and elevation.

In *Alappat*, the Office and BPAI decision addressing the issue on appeal stated that this claim was not statutory subject matter under §101. However, the BPAI decision was overturned by the Federal Circuit.

The reasons given by the BPAI appeared similar to those given in the instant rejection by the Office. The majority decision of the Board had stated that it is proper to treat the above-cited rasterizer claim as if drawn to a method. *See Ex Parte Alappat*, 23 USPQ2d 1340, 1345 (BPAI, 1992). Specifically, the BPAI held that the claims amounted to nothing more than a process claim where each of the steps combine to form a "mathematical

algorithm for computing pixel information." *Alappat* at 1539, quoting *Ex Parte Alappat* at 1345. Further, that "when the claim is viewed without the steps of this mathematical algorithm, no other elements or steps are found." *Ex Parte Alappat* at 1346. The BPAI's reasoning is similar to that of the rejection of the instant claims, where the Office argued that the claimed invention is a "procedure for solving a given type or mathematical problem . . . known as an algorithm" that is "a generalized formulation for problems to solve mathematical problems of calculating multiplication." *Office Action*, p. 2.

The Federal Circuit overturned the decision of the BPAI. The Federal Circuit stated that the BPAI erred in concluding that this rasterizer claim is nothing more than a process claim. *Alappat* at 1540. In deciding that the BPAI erred, the Federal Circuit relied on the language of the claim on its face as well as the claim when read in light of the disclosure of the specification. *Id.* The Federal Circuit also analyzed whether the claimed subject matter as a whole is a disembodied mathematical concept, which in essence represents nothing more than a "law of nature," "natural phenomenon," or "abstract idea." *Id* at 1544.

Applicant sets forth the analysis performed by the Federal Circuit and then applies this analysis to the instant rejection of Claims 1-10.

The Federal Circuit relied on 35 U.S.C. §101, entitled "Inventions patentable," which states that: "Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefore, subject to the conditions and requirements of this title," (*emphasis*

added). Such statute explicitly provides, in unequivocal language, for machines as a statutory category of subject matter for which Applicant is entitled to apply for a patent.

Following this standard, the Federal Circuit's analysis focused on the claim reciting "a rasterizer" and other elements. The Circuit analyzed the language of the claim, concluding that the rasterizer claim recites a machine on its face.

Applicant establishes below that the subject matter recited in independent Claims 1 and 7 recites a machine on its face. Applicant provides independent Claims 1 and 7 below for the convenience of the Board.

Claim 1 recites a computer system, comprising:

- a memory wherein results of processing are stored; and
- a processor that supports SIMD instructions, the processor being configured to perform Montgomery multiplication using SIMD instructions, wherein the Montgomery multiplication has a loop of instructions, and each iteration of the loop involves, excepting copy operations, using no more than eight SIMD instructions and wherein the SIMD instructions comprise two load instructions, one multiply instruction, two add instructions, one copy instruction, one bitwise AND instruction, one store instruction, and one shift instruction.

Claim 7 recites a processing system, comprising:

- a processor having a set of registers, the processor being configured to support SIMD instructions; and
- a set of SIMD instructions, executable by the processor, to perform Montgomery multiplication:
    - montmul(A, B) = rem($(AB - qN)/R$, N), where $q = \text{rem}(AB\,N', R)$.
- where A and B are integers, q is a quotient, N is a modulus, R is an integer that is coprime to modulus N, and N' is an integer such that $N\,N' \equiv 1 \pmod{R}$, wherein the integer B and the modulus N are implemented as arrays, and at least one SIMD instruction is used to update a first array T_1 with multiples of B for computing AB and to update a second array T_2 with multiples of N for computing qN, wherein a first register holds elements of the B and N arrays;

- a second register holds an element of the first array T_1 and an element of the second array T_2; and
- a third register is used to hold results of the first array T_1 being updated with a multiple of B and the second array T_2 being updated with multiples of N.

Claims 1 and 7 recite either a "computer system" or a "processing system." These terms on their face recite a machine—not merely a process or a series of steps performed on a computer as apparently argued by the Office.

The Federal Circuit also studied the disclosure of the specification in deciding whether or not the rasterizer claim constituted a statutory class of subject matter. The Circuit determined that the rasterizer claim recited a machine based on the fact that the disclosure described computer elements that were recited in the claim. The claim recited means-plus-function elements, though this was not dispositive in the Federal Circuit's analysis. Rather, the Federal Circuit relied on the disclosure to show computer elements that may be within the scope of the rasterizer claim. Similarly, exemplary elements are described in the instant specification that, when analyzed as examples of elements recited in the instant claims, show that the instant claims recite a machine and not a process.

Independent Claims 1 and 7 when read in light of the specification, clearly and unequivocally recite machines. It is black letter law that Applicant's claims are to be interpreted in light of the disclosure of the specification. *North Am. Vaccine, Inc. v. American Cyanamid Co.*, 7 F.3d 1571, 1579, 28 USPQ2d 1333, 1339 (Fed. Cir. 1993); *and*

see Miles Lab., Inc. v. Shandon, Inc., 997 F.2d 870, 875, 27 USPQ2d 1123, 1126 (Fed. Cir. 1993).

The specification describes examples of subject matter recited in the claims that are not solely or necessarily a "process" or a "procedure for solving a given type or mathematical problem . . . known as an algorithm" that is "a generalized formulation for problems to solve mathematical problems of calculating multiplication" as argued by the Office. Instead, claims 1 and 7 recite subject matter of one or more machines.

Claim 1 recites "a computer system" comprising "a memory wherein results of processing are stored; and a processor", each of which are described and diagrammed as a machine—not only as a process or a series of steps performed on a computer. Applicant referred the Office and now refers the Board to examples of a computer architecture, memory, and processor in the specification: computer architecture 100 of Figure 1; memory 104 of Figure 1, and microprocessor 102 of Figure 1.

Claim 7 recites "a processing system" comprising "a processor having a set of registers, the processor being configured to support SIMD instructions." Applicant referred the Office and now refers the Board to the above-cited example of a processor, namely: microprocessor 102 of Figure 1 which further includes ALU(s) 106 of Figure 1 and registers 108 of Figure 1. These exemplary elements are shown and described as machines—not necessarily just a "a generalized formulation for problems to solve mathematical problems of calculating multiplication" as argued by the Office.

Not only are these elements described as machines by their usage and diagrammed as machines in the figures, each may also cause results of processing to be stored in a computer's memory. This characteristic, in and of itself, precludes these elements from being solely "a generalized formulation for problems to solve mathematical problems of calculating multiplication" as apparently relied upon by the Office in rejecting Claims 1-10 under §101.

Furthermore, the specification provides the following examples of elements recited in Claims 1 and 7 and described as machines that *perform* an action rather than *being* an action or step:

> Fig. 1 shows a computer architecture 100 that can be configured to implement use of two-way SIMD instructions for Montgomery multiplication. The computer architecture 100 includes a microprocessor 102 and memory 104. The microprocessor 102 has one or more ALUs (Arithmetic Logic Units) 106(1), ..., 106(N) to manage mathematical operations, such as adding and multiplying, and logical operators like OR, AND, XOR, and so forth, as well as right and left shifts. The microprocessor 102 further includes one or more registers 108(1), ..., 108(M) to hold intermediate values and final results produced by the ALUs 106.

Specification, p. 4, ¶ [0013].

Thus, a computer architecture is described in Figure 1 and on at least page 4 of the Specification in language permitting the processor to produce cryptographic results via SIMD instructions for Montgomery multiplication to be: 1) stored in memory; and 2) utilized in cryptographic functions. These characteristics of this exemplary computer system describe a machine—not "a generalized formulation for problems to solve mathematical problems of

calculating multiplication" as argued by the Office. Conversely, a "calculating" cannot be stored in memory—a "calculating" cannot include machine elements. Thus, the characterization put forth by the Office apparently in rejecting Claims 1-10 is inconsistent with the detailed description.

Like the Office in the instant rejection, the BPAI in *Alappat* also argued that the claimed subject matter falls within an exception to §101, namely that it is a "mathematical algorithm." *See Alappat* at 1542. In analyzing the BPAI's position, the Federal Circuit explained the Supreme Court's holdings on mathematical subject matter. The Federal Circuit stated that the Supreme Court "never intended to create an overly broad, fourth category of subject matter excluded from §101." *Alappat* at 1543. Instead, the Federal Circuit explained that this exception to §101 applies to abstract ideas that, in and of themselves, are not entitled to patent protection. The focus in any statutory subject matter analysis must be on the claim as a whole; it is irrelevant that a claim may contain, as part of the whole, subject matter which would not be patentable by itself. *Alappat* at 1543, referring to *Diamond v. Diehr*, 450 U.S. 175, 101 S.Ct. 1048 (1981) *supra*.

The Federal Circuit in *Alappat* concluded that the proper inquiry in dealing with the mathematical subject matter exception of §101 is to determine whether the claimed subject matter as a whole is a disembodied mathematical concept. In essence, it must represent *nothing more* than a "law of nature," "natural phenomenon," or "abstract idea." *See Alappat*

at 1544. If the claim represents more than these, it does not fall within the mathematical subject matter exception to §101.

Applicant submits that the Office has failed to meet the Federal Circuit's standard in *Alappat* in rejecting the instant claims. The Office has failed to show that each of these claims represents nothing more than a law of nature, natural phenomenon, or abstract idea. The Office argues that these claims merely manipulate data or solve a purely mathematical problem without any limitation to a practical application. This is simply not supported by the claims or examples of elements in the claims disclosed in the Specification.

For each of the instant claims, the Office has failed to show that the recited elements fall within the mathematical subject matter exception of §101. For this and the other reasons set forth above, Applicant respectfully submits that the instant claims comply with 35 U.S.C. §101 and requests that the §101 rejections be reversed.

Patent eligible subject matter in claims 15-22 under 35 U.S.C. § 101:

AT&T Corp. v. Excel Communications, Inc.

Applicant respectfully submits that pending claims 15-22 recite methods constituting statutory subject matter under 35 U.S.C. §101. Applicant sets forth the legal standard for a rejection under §101, including *AT&T Corp. v. Excel Communications, Inc.*, 172 F.3d 1352 (1999) addressing a similar rejection under §101 as that of the instant rejection of Claims 15-22. Through analysis of *AT&T*, Applicant will show that the Office's rejection of the instant claims stands in stark disagreement with prevailing law.

It is established law that an abstract idea, by itself, is considered to be unpatentable subject matter under § 101. *See, e.g.*, Id. at 1355 (pointing out that laws of nature, natural phenomena, and abstract ideas have generally been identified by the Supreme Court as unpatentable subject matter). However, if such an idea is taken out of the abstract and employed in some type of process that achieves a "new and useful end", the process is patentable subject matter, even if the idea by itself would not be. *Id.* at 1357. Thus, the relevant inquiry under § 101 becomes -- Is the idea being applied to achieve a useful end? *Id.* If so, then the § 101 threshold is satisfied. *Id.*

In *AT&T*, the invention was designed to operate in a telecommunications system with multiple long-distance service providers. The system contained local exchange carriers ("LECs") and long-distance service (interexchange) carriers ("IXCs"). The LECs provided local telephone service and access to IXCs. Each customer had an LEC for local service and selected an IXC, such as AT & T or Excel, to be its primary long-distance service (interexchange) carrier or PIC. The system involved a three-step process when a caller made a direct-dialed (1+) long-distance telephone call: (1) after the call was transmitted over the LEC's network to a switch, and the LEC identified the caller's PIC, the LEC automatically routed the call to the facilities used by the caller's PIC; (2) the PIC's facilities carried the call to the LEC serving the call recipient; and (3) the call recipient's LEC delivered the call over its local network to the recipient's telephone.

When a caller made a direct-dialed long-distance telephone call, a switch (which could be a switch in the interexchange network) monitored and recorded data related to the call, and generated an "automatic message account" ("AMA") message record. This

contemporaneous message record contained fields of information such as the originating and terminating telephone numbers, and the length of time of the call. These message records were then transmitted from the switch to a message accumulation system for processing and billing.

Because the message records were stored in electronic format, they could be transmitted from one computer system to another and reformatted to ease processing of the information. Thus the carrier's AMA message subsequently was translated into the industry-standard "exchange message interface," forwarded to a rating system, and ultimately forwarded to a billing system in which the data resided until processed to generate, typically, "hard copy" bills which were mailed to subscribers.

The invention at issue in the *AT&T* case called for the addition of a data field into a standard message record to indicate whether a call involved a particular PIC (the "PIC indicator"). This PIC indicator could exist in several forms, such as a code which identified the call recipient's PIC, a flag which showed that the recipient's PIC was or was not a particular IXC, or a flag that identified the recipient's and the caller's PICs as the same IXC. The PIC indicator therefore enabled IXCs to provide differential billing for calls on the basis of the identified PIC.

One of the claims at issue – claim 1-- read as follows:

> A method for use in a telecommunications system in which interexchange calls initiated by each subscriber are automatically routed over the facilities of a particular one of a plurality of interexchange carriers associated with that subscriber, said method comprising the steps of:
>
> generating a message record for an interexchange call between an

> originating subscriber and a terminating subscriber, and
>
> including, in said message record, a primary interexchange carrier (PIC) indicator having a value which is a function of whether or not the interexchange carrier associated with said terminating subscriber is a predetermined one of said interexchange carriers.

In looking at the subject matter of this claim and finding the claim to pass muster under 35 U.S.C. § 101, the Court looked to the *specification* and commented as follows:

> In this case, Excel argues, correctly, that the PIC indicator value is derived using a simple mathematical principle (p and q). But that is not determinative because AT&T does not claim the Boolean principle as such or attempt to forestall its use in any other application. It is clear from the written description of the '184 patent that AT&T is only claiming a process that uses the Boolean principle in order to determine the value of the PIC indicator. The PIC indicator represents information about the call recipient's PIC, a useful, non-abstract result that facilitates differential billing of long-distance calls made by an IXC's subscriber. Because the claimed process applies the Boolean principle to produce a useful, concrete, tangible result without pre-empting other uses of the mathematical principle, on its face the claimed process comfortably falls within the scope of § 101.

The Court looked at the specification and found that the environment and use of the PIC indicator – that of providing differential billing – provided a useful, concrete and tangible result. That result, however, was not specifically recited in the claim. Rather, it was described in the specification.

Likewise, in the present case, the specification provides a description of the utility and tangibility of the recited subject matter, and Applicant has previously amended the claims to include such patentable subject matter, (i.e., a "method for computing Montgomery multiplication, whereby Montgomery multiplication is performed within a cryptographic

function in a computer" and a "method whereby Montgomery multiplication is performed within a cryptographic function in a computer").

Accordingly, in the claims as throughout the Specification, it is evident that the claimed subject matter has a specifically described useful, concrete and tangible result and application.

In view of the above discussion, the Office has failed to show that claims 15-22 present unpatentable subject matter under § 101. Applicant respectfully submits that claims 15-22 comply with the patentability requirements of § 101 and that the § 101 rejections should be withdrawn.

For each of the instant claims, the Office has failed to show that the recited elements fall within the mathematical subject matter exception of §101. For this and the other reasons set forth above, Applicant respectfully submits that the instant comply with 35 U.S.C. §101 and requests that the §101 rejections be withdrawn.

CONCLUSION

The Applicant respectfully considers this application to be in condition for allowance. Appellant submits, as evidenced by the above discussion, Claims 1-4, 7, 8, 10, 12, and 14-22 satisfy the requirements of 35 U.S.C. § 101 and therefore are statutory. Therefore, Appellant respectfully requests that the Board overturn the final rejection and that the Examiner pass this application to allowance.

Dated this 29th day of July, 2008.

Respectfully submitted,

BEATRICE L. KOEMPEL-THOMAS
Attorney for Appellant/Applicant
Registration No. 58,213

KAYLA D. BRANT
Agent for Appellant/Applicant
Registration No. 46,576

LEE & HAYES PLLC
421 W. Riverside Avenue, Suite 500
Spokane, WA 99201
Telephone: (509) 324-9256 (Ext. 259)
Facsimile: (509) 323-8979

VIII. APPENDIX: CLAIMS ON APPEAL

1. (Previously Presented) A computer system comprising:

a memory wherein results of processing are stored; and

a processor that supports SIMD instructions, the processor being configured to perform Montgomery multiplication using SIMD instructions, wherein the Montgomery multiplication has a loop of instructions, and each iteration of the loop involves, excepting copy operations, using no more than eight SIMD instructions and wherein the SIMD instructions comprise two load instructions, one multiply instruction, two add instructions, one copy instruction, one bitwise AND instruction, one store instruction, and one shift instruction.

2. (Original) A computer system as recited in claim **1**, wherein the processor is executing a cryptographic function and the Montgomery multiplication is used to compute exponentiations in the cryptographic function.

3. (Original) A computer system as recited in claim **1**, wherein the processor maintains two arrays to hold intermediate computations from the Montgomery multiplication, and the SIMD instructions are used to simultaneously update the two arrays.

4. (Original) A computer system as recited in claim **1**, wherein the Montgomery multiplication involves a first multiplication of an input array and a second multiplication of a modulus array, and the SIMD instructions are used to perform simultaneously the first and second multiplications.

5. (Canceled)

6. (Canceled)

7. (Previously Presented) A processing system comprising:

a processor having a set of registers, the processor being configured to support SIMD instructions; and

a set of SIMD instructions, executable by the processor, to perform Montgomery multiplication:

$$\text{montmul}(A, B) = \text{rem}(\,(AB - qN)/R, N), \quad \text{where} \quad q = \text{rem}(AB\,N', R).$$

where A and B are integers, q is a quotient, N is a modulus, R is an integer that is coprime to modulus N, and N' is an integer such that $N\,N' \equiv 1 \pmod{R}$, wherein the integer B and the modulus N are implemented as arrays, and at least one SIMD instruction is used to

update a first array T_1 with multiples of B for computing AB and to update a second array T_2 with multiples of N for computing qN, wherein a first register holds elements of the B and N arrays;

a second register holds an element of the first array T_1 and an element of the second array T_2; and

a third register is used to hold results of the first array T_1 being updated with a multiple of B and the second array T_2 being updated with multiples of N.

8. (Original) A processing system as recited in claim 7, wherein the SIMD instructions comprise a single SIMD instruction that simultaneously performs parts of the multiplications AB and qN.

9. (Canceled)

10. (Original) A processing system as recited in claim 9, wherein a single SIMD instruction is used to update the first array T_1 and the second array T_2 simultaneously.

11. (Canceled)

12. (Previously Presented) A computer readable medium comprising computer-executable SIMD instructions that, when executed, direct a processor to perform Montgomery multiplication, the instructions comprising:

a first SIMD instruction to load elements of arrays B and N into a first register;

a second SIMD instruction to load elements of arrays T_1 and T_2 into a second register;

a third SIMD instruction to multiply an element in the array B by a first multiple and an element in the array N by a second multiple;

fourth and fifth SIMD instructions to add results of the third SIMD instruction to the array elements loaded by the second SIMD instruction and to any carries saved from a previous iteration;

sixth and seventh SIMD instructions to separate each output of the fifth SIMD instruction into two reduced size results, one that fits into the arrays T_1 and T_2 and another that represents a carry for a next iteration;

an eighth SIMD instruction to update an element of array T_1 and an element of array T_2, in memory; and

an instruction to store the result of the final iteration.

13. (Canceled)

14. (Original) A computer readable medium as recited in claim 0, wherein the SIMD instructions comprise SSE2 instructions.

15. (Previously Presented) A method for computing Montgomery multiplication, whereby Montgomery multiplication is performed within a cryptographic function in a computer, the method comprising:

$$\text{montmul}(A, B) = \text{rem}(\ (AB - qN)/R, N), \quad \text{where} \quad q = \text{rem}(AB\ N', R).$$

where A and B are integers, q is a quotient, N is a modulus, R is an integer that is coprime to modulus N, and N' is an integer such that $N\,N' \equiv 1 \pmod{R}$, the method comprising:

iteratively performing, for each digit of integer A from right to left:

with array T_1 being updated by a product of input B times the digit of integer A, determining what multiple of modulus N allows the updated arrays T_1, T_2 to end with the same digit;

multiplying the input B by the digit of integer A and multiplying the modulus N by the determined multiple; and

updating the arrays T_1, T_2

storing the result of the final iteration.

16. (Original) A method as recited in claim 15, wherein the performing comprises using SIMD instructions.

17. (Original) A method as recited in claim 15, wherein the multiplying is performed by a single SIMD instruction.

18. (Original) A method as recited in claim 15, further comprising initializing the arrays T_1, T_2 and the modulus N prior to said performing.

19. (Original) One or more computer readable media storing computer executable instructions that, when executed, perform the method as recited in claim 15.

20. (Previously Presented) A method whereby Montgomery multiplication is performed within a cryptographic function in a computer, the method comprising:

initializing a set of registers with values used in performing Montgomery multiplication;

computing the Montgomery multiplication with SIMD instructions on the values stored in the registers, wherein the Montgomery multiplication has a loop of instructions, and each iteration of the loop is performed using not more than nine SIMD instructions wherein

the nine SIMD instructions comprise:

two load instructions;

one multiply instruction;

two add instructions;

one copy instruction;

one bitwise AND instruction;

one store instruction; and

one shift instruction; and

storing the result of the final iteration of the loop.

21. (Original) A method as recited in claim 20, wherein the computing comprises using the Montgomery multiplication to compute exponentiations in a cryptographic function.

22. (Original) A method as recited in claim 20, wherein the computing comprises using SSE2 instructions.

23. (Canceled)

24. (Canceled)

IX. APPENDIX: EVIDENCE

None.

X. APPENDIX: RELATED PROCEEDINGS

None.